

# Simple and Efficient Key Management Method for Hierarchical Wireless Sensor Networks

Mohammed A. Altaha<sup>1</sup>, Wisam Mahmood Lafta<sup>2</sup>, Ghazwan Abdulnabi Al Ali<sup>3</sup>, Ahmed Adil Alkadhma<sup>4</sup>

<sup>1</sup>Department of Veterinary Public Health, College of Veterinary, University of Basrah, Basrah, Iraq

<sup>2</sup>Department of Computer Science, University of Technology, Baghdad, Iraq

<sup>3</sup>Department of Computer Science (Educational Science), University of Basrah, Basrah, Iraq

<sup>4</sup>Department of English, College of Education for Human Sciences, University of Basrah, Basrah, Iraq

---

## Article Info

### Article history:

Received Jul 29, 2024

Revised Sep 26, 2024

Accepted Oct 8, 2024

---

### Keywords:

Hash function

Key management

XOR operation

Wireless sensor network.

---

## ABSTRACT

Security is an important consideration for Wireless Sensor Networks (WSNs), and key management plays a pivotal role in facilitating safe communication and data transfer. Key management must be designed with the constraints of these networks in mind, which include limited computation capabilities, memory, and energy. Achieving secure and efficient communication in large-scale WSNs is a significant challenge. In this paper, we propose a simple key management method for securing hierarchical WSNs, which employs only a few hash functions and XOR operations to derive shared keys. Its simplicity makes optimal use of resources and offers an efficient approach to establishing keys for sensor nodes. Simulation results demonstrate that the proposed scheme reduces energy consumption by 15% and decreases the key establishment time by 20% compared to existing methods such as LKMS, while maintaining strong security with low computational and communication costs, which are crucial considerations for WSNs.

Copyright © 2024 Institute of Advanced Engineering and Science.  
All rights reserved.

---

## Corresponding Author:

Mohammed A. Altaha

Department of Veterinary Public Health, College of Veterinary, University of Basrah

Basrah, Iraq

Email: mohammed.altaha@uobasrah.edu.iq

---

## 1. INTRODUCTION

Wireless Sensor Networks (WSNs) are generally composed of tiny sensor nodes with limited energy, memory, and computing abilities [1]. These networks use small batteries as power for the nodes. WSNs are used in various areas, including in the military and commercial sectors, healthcare applications, environmental monitoring, traffic control, and industrial applications [2–6]. However, WSNs have some inherent characteristics that make them highly susceptible to various security threats, including distributed deployment, a dynamic topology, limited resources, and susceptibility to physical attacks. Since the data that are transmitted from the sensor nodes may contain highly sensitive and confidential information, the confidentiality, integrity and availability of the data become prime considerations when designing a WSNs [7].

WSNs can be broadly classified into two main categories, depending on whether they are flat or hierarchical in nature. In a flat WSN, each node at the sensor level has the same features in terms of data gathering and transfer over that particular network. This type of network structure is suitable for small-scale networks, as these are usually easy to manage and require fewer resources. In contrast, hierarchical WSNs impose a certain degree of hierarchy on the sensor nodes depending on their capabilities. In this framework,

lower-level nodes, which have severely restricted capabilities in terms of storage, computation, and energy, reside at the base of the hierarchy. Above them are cluster heads, which have enhanced capabilities compared to the sensor nodes. At the top of the hierarchy is a mobile sink or base station, which has the most extensive resources, and which collects and analyzes the data transmitted by the cluster heads. The use of a hierarchical structure in a WSNs allows for efficient data aggregation, scalable network management, and improved energy utilization [8, 9].

The area of network security encompasses a collection of services, mechanisms and general policies that protect a network from unauthorized access and potential attacks [10]. In general, WSNs security requires the use of certain methods and tools [11]. Although the introduction of security to such a network is very advantageous, because it can avoid any possible communication problems, these security processes may cause computational and communication overload, causing the system to demand more energy and memory [12].

In order to tackle the abovementioned security concerns, several cryptographic methods have been proposed, and in some of these, key management plays a very important role. Key management is the process of generating, distributing, updating, and revoking the keys that are used for encryption of messages that are passed from one node to the other in the network. Thus, good schemes for key management are critical for establishing secure and dependable operations in WSNs [13]. The objective of key management in WSNs is to employ reliable and efficient methods for the development of secure channels between the sensor nodes and the base station for communication. This includes ones that address design and implementation of efficient cryptographic algorithms and protocol that can be used by resource-limited sensor nodes. Furthermore, key management schemes should be secure from node compromise, physical attacks, and key compromise [14, 15].

This paper therefore presents an efficient, simple key management method for hierarchical WSN security. The generation of shared keys uses simple hashing functions and XOR operations, with a focus on low resource consumption and fast key creation among clusters. First, a large pool of keys are generated by the base station, and a square key matrix is generated. Then, each sensor node is pre-loaded with the  $i$ <sup>th</sup> row and column from the matrix key. These keys are used to generate the shared keys among the nodes, which are utilized to protect data transmissions between sensor nodes within the network.

The rest of this paper is organized as follows. A comprehensive survey of related work is presented in Section 2. Section 3 introduces the network model, and Section 4 describes the proposed method. Section 5 explains the experimental study and presents the results, and Section 6 contains the conclusion.

## 2. RELATED WORK

In recent years, a considerable amount of research has been conducted on developing key management techniques that are tailored to the unique characteristics and constraints of WSNs. These techniques aim to enhance the security of WSNs by providing mechanisms for secure key establishment, key storage, key revocation, and key updating. A given approach to key management cannot be used for all applications, due to the many types of WSN applications and their various security requirements; it is therefore crucial to select the right key management technique for each application. In this section, we analyze the research background, and introduce and explain a number of important management techniques.

Wuu *et al.* [16] proposed a Quorum-Based Key Management Scheme for securing WSNs, in which each sensor was pre-loaded with a subset of the quorum system. When two nodes need to communicate, they first exchange IDs, and then calculate the shared key from the assigned subset. Daghighi *et al.* [17] proposed a hierarchical key management scheme with a re-keying strategy for mobility nodes in order to secure group communications, called HISCOM. HISCOM used Traffic Encryption Keys (TEKs), which were shared among authorized nodes for encrypting and decrypting group-intended data, and included a two-tier hierarchical approach with a Domain Key Manager (DKM) and Area Key Managers (AKM) to manage group communication. Anzani *et al.* [18] proposed a scheme called Merging Hybrid Symmetric Design (MGHS), which was a hybrid key pre-distribution technique with a symmetric design as a key management strategy for WSNs. The proposed method used key rings, which were generated by combining blocks with a symmetric design. A parameter  $d$  was used for determining the number of merged blocks to generate the key-rings. Before deployment, a number of key-rings were generated in the base station and pre-loaded inside node's memory. The key rings were used to provide secure communication between two sensor nodes, and each node uses a unique key that was shared between the key rings. Altaha and Muhajjar [19] proposed a Lightweight Key Management Scheme (LKMS) for securing hierarchical WSNs. The proposed method employ a symmetric key which consist of only a hash

function and a XOR operation. Yousefpour *et al.* [20] propose a dynamic key management method for hierarchical WSNs by employing mamdani fuzzy inference mechanism for selecting the best cluster-head when adding new nodes to the network and for generating the path key. The proposed method has four type of keys: a route key, pair key, cluster key, and a private key. Singh *et al.* [21] proposed a multi-level key management for securing a hierarchical WSNs. The proposed method used Unmanned Aerial Vehicle (UAV) as a secured third-party to generate a session keys between the sensor node. The proposed method used only a hash function and XOR operation to generate the session keys. Cheng *et al.* [22] proposed a scheme for group key generation and authentication by employing an asymmetric bivariate polynomial to secure the WSNs. Each node combined it input with pairwise shared keys with other nodes and broadcasts the encrypted value into a channel. Then each node can compute the group key after collecting all released values. Helali *et al.* [23] proposed a Key Management method using a Pool-Hash (KMPH) to secure the WSNs. The proposed method introduces a probabilistic key pre-distribution strategy that seeks to generate a collection of keys including both the hashed keys and the original keys. KMPH consists of three essential stages in managing keys: shared key discovery, key pre-distribution, and path key setup. KMPH facilitates the dissemination of the secured keys among the sensor nodes. During the key discovery phase and the path key setup, new session keys were calculated and transferred between nodes. Armogam and Seshasayanam [24] proposes an Adaptive Multilevel Location-based Key Management System (AML-KBS). The proposed method dynamically generated the keys and shared with each nodes in the networks. AML-KBS provide different types of keys: a sensor key which was shared between the base station and the sensor node, an asymmetric key set, a neighborhood key which was exchanged by physically nearby nodes to authenticate their identities and to safeguard their data from malicious nodes, and a sector key that was shared among nodes of a particular sector. AML-KBS utilizes a location-based approach for key management, enabling the identification of attackers based on their geographical location. Kumar *et al.* [25] proposed a Scalable and Storage Efficient Key Management Scheme (SSEKMS). Three kinds of keys used in the proposed method namely: pairwise key, network key, and cluster key. The pairwise key was used to ensure the security of the communication between two nodes. The network key was distributed across all the nodes. The cluster key was distributed across all the nodes within the same cluster. SSEKMS was dynamic key management method and can supports adding new nodes and refreshing the keys as needed. Ahlawat and Dave [26] proposed a method for securing path key establishment based on random key management. The proposed method used an Attack Matrix (AM) which was designed to determine the sensor nodes' attack coefficient when they were placed in the sensor field. By avoiding the nodes with the highest attack coefficient value, path key exposure was minimized.

### 3. NETWORK MODEL

The key management method presented in this paper utilizes a hierarchical WSN topology consisting of many sensor nodes, cluster heads, and a trusted base station, as shown in Figure 1. The network is divided into several clusters, and all nodes are static. Each cluster has several sensor nodes and a cluster head. The cluster head nodes act as intermediaries, and are responsible for aggregating and processing the collected data before transmitting them to the base station. The enhanced capabilities of the cluster head nodes enable efficient data management and aggregation, thereby contributing to improved network performance [27].

### 4. PROPOSED METHOD

A hierarchical key management scheme is proposed, which consists of a base station, multiple cluster heads representing powerful nodes, and ordinary sensor nodes.

#### 4.1. Pre-distribution phase

Before deployment, confidential information is generated by the base station and shared among the sensor nodes.

Inspired by [16], a large pool of keys is generated by the base station and then used to generate a square key matrix  $K$  with size  $nn$ , where  $n$  is the number of nodes to be deployed (see Figure 2). Each sensor node is pre-loaded with the  $i_{th}$  row and  $i_{th}$  column of the key matrix  $K$ . For example, node  $A$  is uploaded with a row index of three and a corresponding column with an index of three. For each node, a unique identifier ( $ID_i$ ) will be uploaded to the node memory. Algorithm 1 summarizes the steps applied in the pre-deployment phase.

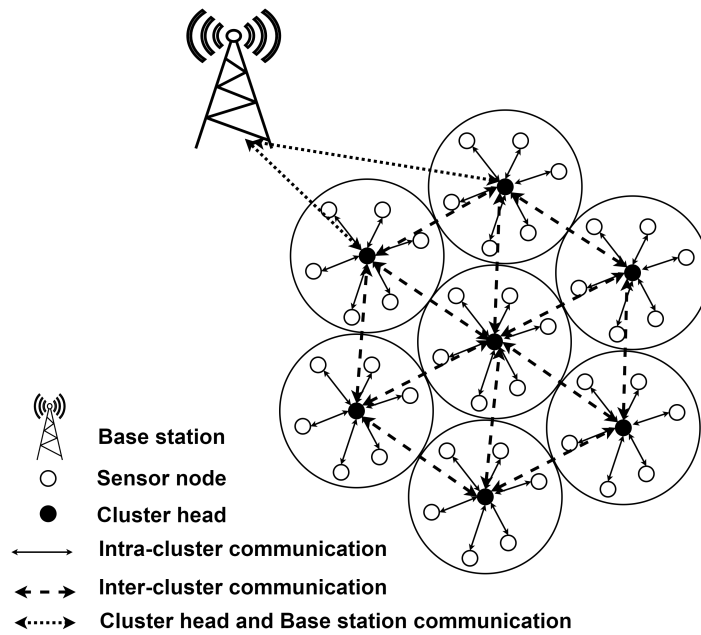


Figure 1. The network model

$K_{1,1}$	$K_{1,2}$	$K_{1,3}$	...	$K_{1,n}$
$K_{2,1}$	$K_{2,2}$	$K_{2,3}$	...	$K_{2,n}$
$K_{3,1}$	$K_{3,2}$	$K_{3,3}$	...	$K_{3,n}$
...	...	...	...	...
$K_{n,1}$	$K_{n,2}$	$K_{n,3}$	...	$K_{n,n}$

Figure 2. The key matrix

#### 4.2. key management phase

After deployment, the network first enters the clustering phase. Each cluster head sends a *HELLOMESSAGE* containing its *ID*, and each node connects with the nearest cluster head. Each sensor node can receive signals from several cluster heads, and selects a cluster head based on the optimal signal intensity. Following this, the key management phase begins. To calculate the shared key, each node discovers its neighboring nodes in the same cluster.

Suppose *A* and *B* are two nodes that want to communicate with each other. Node *A* sends a message to node *B* containing its *ID*. Node *B* calculates  $M_1$ , which is the hash value of the pre-loaded key value equal to the indexed row of Node *B* (which is equal to the *ID* of Node *A*), as shown in Equation 1.

$$M_1 = hash(B_{row}[A_{ID}]) \tag{1}$$

Node *B* then selects a random number  $r_B$  and encrypts it with  $M_1$  as the key, using Equation 2.

$$R_B = E_{M_1}(r_B \oplus B_{row}[A_{ID}]) \tag{2}$$

Then, node *B* sends  $[M_1, R_B, B_{ID}]$  to node *A*. Node *A* calculates  $M_1'$  using Equation 3 and compares the value with  $M_1$ . If  $M_1' = M_1$  then it continues to extract  $r_B'$ , otherwise, the session will terminate.

$$M_1' = hash(A_{column}[B_{ID}]) \tag{3}$$

**Algorithm 1** Pre-deployment Phase**Before deployment the sensor nodes:**

Generate a large pool of keys at the base station.

Generate a square key matrix  $K$  of size  $n \times n$ .

Pre-load each sensor node  $i$  with the  $i_{th}$  row and  $i_{th}$  column of the matrix  $K$ .

Assign each node a unique identifier  $ID_i$ .

$$r_{B'} = D_{M_1'}(R_B) \oplus A_{column}[B_{ID}] \quad (4)$$

Node  $A$  then calculates  $M_2$  based on Equation 1, using the indexed  $B_{ID}$  row of node  $A$ .

$$M_2 = hash(A_{row}[B_{ID}]) \quad (5)$$

Node  $A$  selects a random value and calculates the  $R_A$  in the same way, using Equation 2, which is encrypted using  $M_2$  as the encryption key.

$$R_A = E_{M_2}(r_A \oplus A_{row}[B_{ID}]) \quad (6)$$

Node  $A$  sends  $[M_2, R_A, A_{ID}]$  to node  $B$ , which then calculates  $M_2'$  using Equation 7 and compares the value with  $M_2$ . If  $M_2' = M_2$  it continues and extracts  $r_{A'}$ , otherwise, the session will terminate.

$$M_2' = hash(B_{column}[A_{ID}]) \quad (7)$$

$$r_{A'} = D_{M_2'}(R_A) \oplus B_{column}[A_{ID}] \quad (8)$$

Finally, Nodes  $A$  and  $B$  calculate the shared key between them as follows:

$$K_{AB} = hash(r_{A'} \oplus r_{B'}) \quad (9)$$

The shared key  $K_{AB}$  is used to secure the communication between the two nodes (node  $A$  and node  $B$ ).

Each node carries out these steps to calculate the shared keys with the corresponding neighbor nodes, and uses them to secure their communication. Algorithm 2 and Figure 3 illustrate the process of shared key generation.

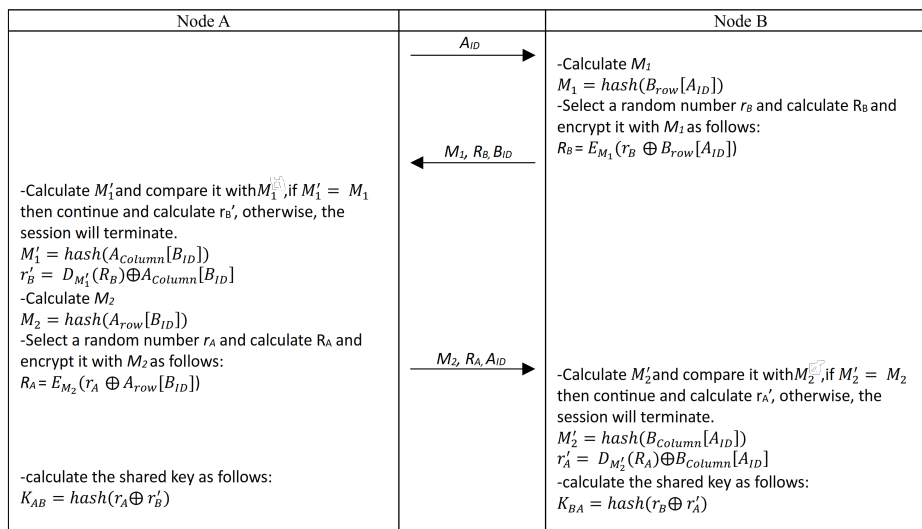


Figure 3. Shared key generation.

**Algorithm 2** Shared Key generation

Each node discovers its neighbor nodes within the same cluster.

**Nodes A and B wish to communicate**

**Step 1: Node A to Node B**

Node A sends its ID to Node B.

Node B receives Node  $A_{ID}$ , and do the following:

- calculates  $M_1$
- selects a random number  $r_B$
- calculates  $R_B$
- sends  $[M_1, R_B, B_{ID}]$  to Node A.

**Step 2: Node A Verification**

Node A calculates  $M'_1$

**if**  $M'_1 = M_1$  **then**

Node B is authenticated. Node A :

- calculates  $M_2$
- selects a random number  $r_A$
- calculates  $R_A$
- sends  $[M_2, R_A, A_{ID}]$  to Node B.

**else**

Terminates session.

**end if**

**Step 3: Node B Verification**

Node B calculates  $M'_2$

**if**  $M'_2 = M_2$  **then**

Node A is authenticated

**else**

Terminates session.

**end if**

**Step 4: Shared Key Calculation**

Both nodes calculate the shared key  $K_{AB}$

$K_{AB}$  is used to secure communications between Nodes A and B.

#### 4.3. Key update

For security reasons, the keys need to be updated periodically or on demand, in order to remove compromised sensor nodes and to protect against eavesdropping attacks. After a certain period, each node updates the shared keys with their corresponding neighbors by taking the hash value of the current keys, as shown in Equation 10.

$$K'_{AB} = hash(K_{AB}) \quad (10)$$

#### 4.4. Add new nodes

Due to its finite energy, a sensor node will eventually die, and a new node will need to be deployed in place of a dead one. Before deployment, the new node will be pre-loaded with its  $ID$  and a secret key,  $K_N$ . After deployment, the new node sends a *HELLOMESSAGE* to the nearest cluster head, encrypted with the pre-loaded secret key, to join a cluster. The cluster head sends the new node  $ID$  to the base station. Then, the base station replays the new node secret key encrypted by the shared key between the base station and cluster head. Following this, the cluster head decrypts the *HELLOMESSAGE* using the key received from the base station to authenticate the new node, and calculates the shared key between the new node and the cluster members using Equation 11.

$$K_{AN} = hash(K_N \oplus K_{AC}) \quad (11)$$

where  $K_{AN}$  represents the new shared key between Node A and the new node;  $K_N$  is the secret key of the new node; and  $K_{AC}$  is the shared key between Node A and the cluster head.

The cluster head then sends the shared key to Node  $A$  and the new node (it is sent to node  $A$  encrypted with the  $K_N$  key, and to the new node encrypted with the  $K_N$  key). The cluster head calculates the shared key between the cluster members and the new node.

---

**Algorithm 3** Adding a new node to the WSN
 

---

The new node is pre-loaded with its ID and a secret key  $K_N$ .

The new node is deployed in place of a dead one.

The new node sends a HELLO MESSAGE to the nearest cluster head, encrypted with  $K_N$ .

The cluster head receives the HELLO MESSAGE.

The cluster head sends the new node's ID to the base station.

The base station sends the new node's secret key  $K_N$  to the cluster head.

The cluster head decrypts the HELLO MESSAGE using the key received from the base station and authenticates the new node.

**for** each  $node_i$  in the cluster **do**

cluster head calculates the shared key  $K_{iN}$  between  $node_i$  and the new node  $N$ :

$$K_{iN} = \text{hash}(K_N \oplus K_{iC})$$

cluster head sends the shared key  $K_{iN}$ :

- to  $node_i$  encrypted with the  $K_{iC}$  key.

- to the new node encrypted with the  $K_N$  key.

**end for**

---

## 5. SIMULATION AND RESULTS

### 5.1. Performance analysis

To evaluate the effectiveness of our method, experiments were conducted using a simulation in MATLAB, where 100, 500, and 1000 nodes were randomly deployed over sensing areas of  $100m^2$ ,  $200m^2$ , and  $250m^2$  respectively. Table ?? summarizes the simulation parameters. Two primary metrics were used to evaluate the proposed method: energy consumption and time consumption. The proposed method was compared against two existing key management schemes, LKMS [19] and Singh's method [21].

Table 1. Simulation parameters

Parameter	Value
Network size	$100m^2, 200m^2, 250m^2$
Number of nodes	100, 500, 1000
Initial energy	$0.5j$
Node placement	Randomly
Amplifier transmitting energy	$100nj/bit/m^2$
Electronics energy	$50nj/bit$
Size of Data Packet	2Kbit
NO. of Rounds	1000

Energy efficiency is a crucial concern in key management approaches, due to the energy constraints on WSNs. Figure 4 summarizes the energy consumption of the proposed method compared with the alternative key management methods. The results indicate that the proposed method significantly reduces energy consumption when generating the shared keys compared to the other two methods. This improvement is attributed to the efficient use of pre-loaded keys and the streamlined process for establishing shared keys between nodes. The reduction in energy consumption enhances the overall lifespan of the network, meaning that the proposed method is more suitable for practical WSN deployments where energy efficiency is crucial.

Figure 5 summarizes the time-consuming process of generating shared keys to secure communication and compares the results with those of the other methods. One of the vital metrics that affect the performance of WSNs is the time consumed in management. Key management is important when securing the exchanges made over a network, but it needs to be done efficiently to allow ample time for a network that might be affected by delays due to key management. Considerable time is spent on key management, which leads to latency of

time-sensitive information and synchronization of the sensor nodes.

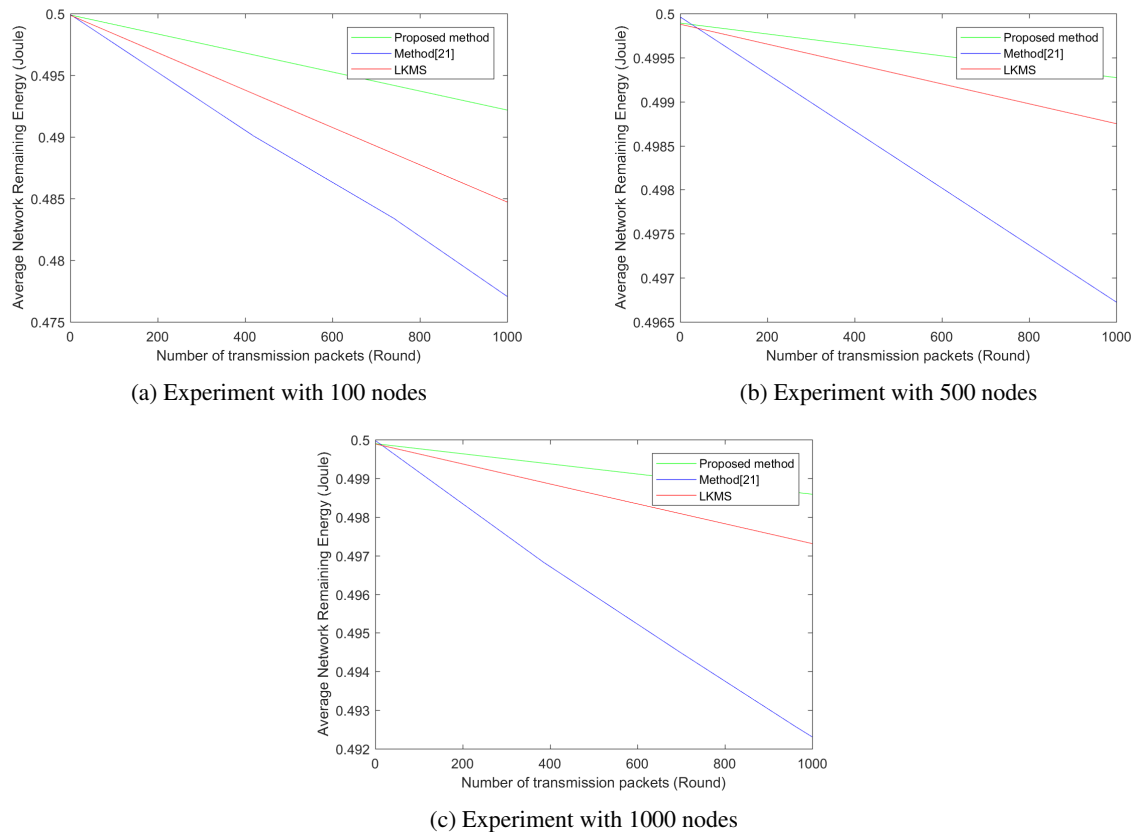


Figure 4. Energy consumption of the proposed method compared with other methods

The evaluation results show that the proposed key management method achieves better performance than the previous schemes, LKMS [19] and Singh's method [21], in terms of both energy consumption and time consumption. With regard to energy, the proposed method achieves a substantial reduction across all scenarios, with deployments of 100, 500, and 1000 nodes in the specified sensing areas. This efficiency is mainly attributed to the reduction of the communication overhead and also the key establishment which employs the use of pre-stored keys and efficient use of hash functions and the XOR operations. In the aspect of time consumption, the proposed method also revealed significant changes. Compared to other methods, the amount of time required to perform key management operations, such as generation, distribution, and establishment, was considerably reduced. Therefore, the proposed method is able to address the issues of energy consumption and time required for the key management while providing faster and more effective secure communication to improve the performance and robustness of WSNs.

The proposed method improves energy consumption and time efficiency primarily by minimizing computational complexity. Unlike LKMS [19], which utilizes symmetric key cryptography but involves additional key storage and management overhead for each sensor node, our method leverages a pre-distributed key matrix and uses only hash functions and XOR operations. This significantly reduces the computational burden on individual sensor nodes. For instance, in our simulation, the energy consumption for 100 sensor nodes was reduced by approximately 15% compared to LKMS [19], due to the elimination of costly cryptographic operations such as modular exponentiations.

Singh's method [21], which employs UAVs for key management, introduces higher communication delays, as it relies on centralized key distribution through the UAVs. In contrast, our method avoids such delays by allowing cluster heads to handle key management locally, resulting in a 20% reduction in key establishment time. In addition, Singh's method suffers from higher energy consumption due to frequent UAV interactions, while our method reduces communication overhead by using cluster heads, thus enhancing network longevity.



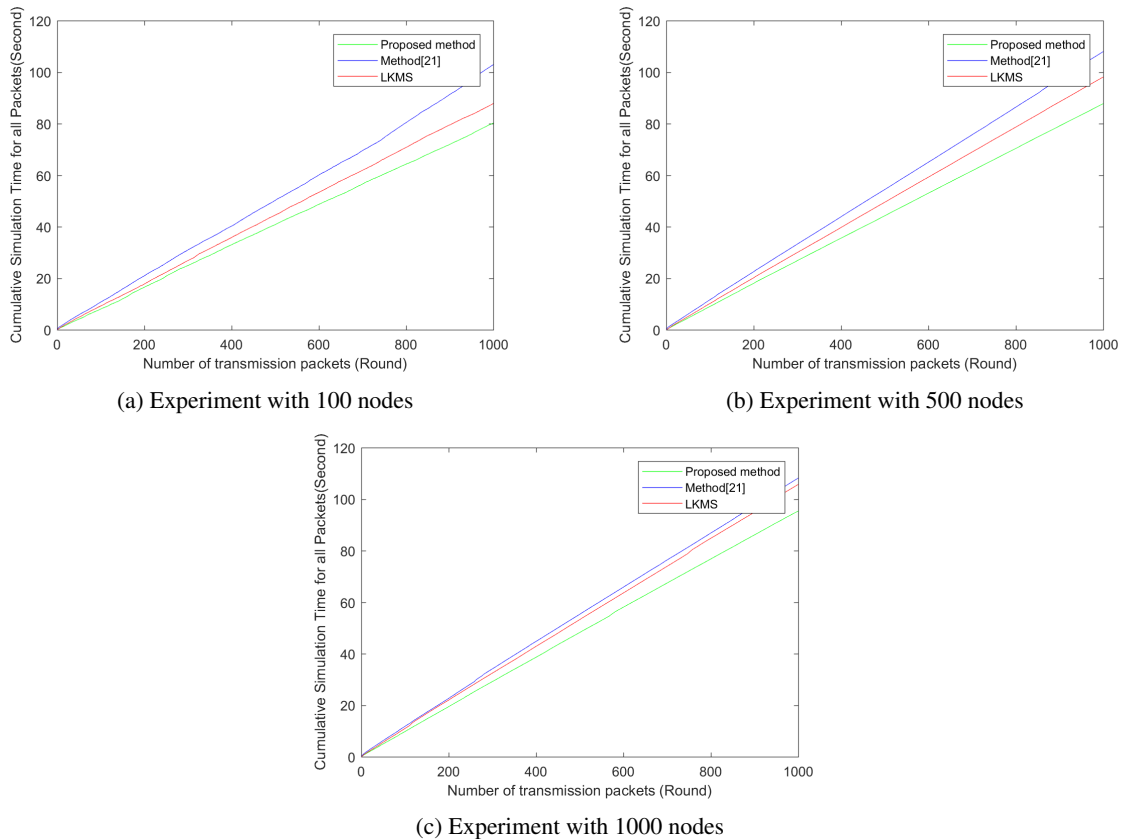


Figure 5. Time consumption of the proposed method compared with other methods

## 5.2. Security analysis

### 5.2.1. Resistance to the physical node capture attack

The proposed key management system provides strong security in the case of a physical node capture, in which an attacker gains direct access to a sensor node. The shared session keys, which are generated from random integers and hash functions, are unpredictable and cannot be used again for other sessions, even if the attacker manages to retrieve the pre-loaded keys on the captured node. The key update mechanism further limits the damage by periodically refreshing the keys, thus ensuring that the compromised node's keys become obsolete after a certain period. Both the Singh's method [21] and LKMS [19] face challenges when nodes are physically captured and keys are exposed. Our proposed scheme reduces the impact of node capture by periodically updating keys using lightweight hashing and random values. As a result, even if a node is compromised, future communications will not be affected after a key update.

### 5.2.2. Resistance to the replay attack

Our key management method for hierarchical WSNs is designed to resist replay attacks, a common threat in WSNs in which malicious nodes capture and retransmit valid data packets to disrupt network operations. To mitigate this risk, our method incorporates a mechanism where each communication session between two sensor nodes involves the generation of unique, session-specific keys. The session keys are generated using random numbers selected by both communicating nodes during the key setup procedure, together with pre-distributed keys and hash functions. Due to the variability of random values in each session, even if an attacker intercepts a message, the encryption keys used in subsequent connections will differ, making the replayed message ineffective. In addition, each message exchange is verified through hash comparisons, thus ensuring that the session keys are valid and preventing any unauthorized access or replayed messages from being accepted by legitimate nodes. Unlike Singh's method [21], which is vulnerable to replay attacks due to its static key distribution model, our method utilizes session-specific keys and timestamps, ensuring that even if an attacker captures a message, it cannot be reused. This makes our method more resilient to replay attacks.

### 5.2.3. Resistance to the denial of service attacks

To counter Denial of Service (DoS) attacks, which aim to exhaust the resources of a sensor node by overwhelming it with unnecessary requests, the proposed method is designed to minimize resource consumption during key management. By utilizing lightweight cryptographic operations such as hash functions and XOR, the computational overhead is kept low, reducing the opportunity for attackers to drain node resources. Additionally, the clustering mechanism helps distribute processing tasks, preventing any single node from becoming overwhelmed. The periodic key update mechanism also ensures that malicious nodes cannot continuously exploit a previously compromised key, further limiting the impact of DoS attacks.

### 5.2.4. Resistance to the node compromising attacks

The proposed method includes several safeguards against node compromise attacks, where an attacker takes control of a sensor node to access the network. By using a hierarchical structure with cluster heads, the impact of a compromised node is confined to its local cluster, limiting the extent of the damage. The use of unique identifiers for each node, along with session-specific shared keys, ensures that a compromised node cannot reuse or distribute its keys to other nodes. Furthermore, the system's key update process ensures that any compromised keys are replaced at regular intervals, minimizing the time window in which an attacker can exploit the compromised node.

### 5.2.5. Resistance to the de-synchronization attack

De-synchronization attacks occur when an adversary repeatedly forges or delays messages between sensor nodes, causing them to lose synchronization and disrupt communication. The proposed key management method mitigates this by relying on periodic key updates and session-specific key exchanges, ensuring that nodes remain synchronized in their communication. Each session uses a fresh shared key derived from pre-distributed keys, random values, and hash functions, preventing attackers from forging outdated messages. Additionally, the mutual verification of keys between nodes ensures that only valid, up-to-date communication occurs, making it difficult for attackers to de-synchronize nodes.

## 6. CONCLUSION

This paper proposes a key management method for securing hierarchical WSNs, which can be considered a simple and effective way of enhancing energy efficiency and time consumption. The proposed method minimizes the amount of information exchanged and the calculations needed for key management by using hash functions and XOR operations. The results indicate that the proposed method provides better performance compared to existing schemes by consuming less energy and less time. These improvements help in increasing the life span of the network as well as its responsiveness, which are critical factors for the practical deployment of WSNs. Overall, the proposed method provides a robust and efficient framework for secure communication in WSNs, making it a viable option for various applications that require reliable and energy-efficient network performance.

## REFERENCES




- [1] T. Qiu, N. Chen, K. Li, D. Qiao, and Z. Fu, "Heterogeneous ad hoc networks: Architectures, advances and challenges," *Ad Hoc Networks*, vol. 55, pp. 143–152, 2017.
- [2] M. M. Afsar and M.-H. Tayarani-N, "Clustering in sensor networks: A literature survey," *Journal of Network and Computer applications*, vol. 46, pp. 198–226, 2014.
- [3] M. A. Altaha, A. A. Alkadhawee, and W. M. Lafta, "Uneven clustering and fuzzy logic based energy-efficient wireless sensor networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 2, pp. 1011–1019, 2022.
- [4] X. He, M. Niedermeier, and H. De Meer, "Dynamic key management in wireless sensor networks: A survey," *Journal of network and computer applications*, vol. 36, no. 2, pp. 611–622, 2013.
- [5] S. Abdollahzadeh and N. J. Navimipour, "Deployment strategies in the wireless sensor network: A comprehensive review," *Computer Communications*, vol. 91, pp. 1–16, 2016.

- [6] A. A. Alkadhawee, M. A. Altaha, and W. M. Lafta, "Unequal clustering algorithm with ida\* multi-hop routing to prevent hot spot problem in wsns," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, pp. 445–453, 2020.
- [7] R. Di Pietro, S. Guarino, N. V. Verde, and J. Domingo-Ferrer, "Security in wireless ad-hoc networks—a survey," *Computer Communications*, vol. 51, pp. 1–20, 2014.
- [8] A. Albakri, L. Harn, and S. Song, "Hierarchical key management scheme with probabilistic security in a wireless sensor network (wsn)," *Security and communication networks*, vol. 2019, no. 1, p. 3950129, 2019.
- [9] S. P. Singh and S. C. Sharma, "A survey on cluster based routing protocols in wireless sensor networks," *Procedia computer science*, vol. 45, pp. 687–695, 2015.
- [10] L. Chen, J. Ji, and Z. Zhang, *Wireless network security*. Springer, 2013.
- [11] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in internet-of-things," *IEEE Internet of things Journal*, vol. 4, no. 5, pp. 1250–1258, 2017.
- [12] M. A. El-Bendary, *Developing security tools of WSN and WBAN networks applications*. Springer, 2015.
- [13] J.-M. Huang, S.-B. Yang, and C.-L. Dai, "An efficient key management scheme for data-centric storage wireless sensor networks," *IERI Procedia*, vol. 4, pp. 25–31, 2013.
- [14] G.-C. Yang, S.-I. Ao, and L. An, *IAENG transactions on engineering technologies*. Springer, 2015.
- [15] Y. Wang and J. Zhao, "Key management scheme for wireless sensor networks," in *Wireless Internet: 10th International Conference, WiCON 2017, Tianjin, China, December 16-17, 2017, Proceedings 10*. Springer, 2018, pp. 272–283.
- [16] L.-C. Wu, C.-H. Hung, and C.-M. Chang, "Quorum-based key management scheme in wireless sensor networks," in *Proceedings of the 6th international conference on ubiquitous information management and communication*, 2012, pp. 1–6.
- [17] B. Daghighi, M. L. Mat Kiah, S. Iqbal, M. H. U. Rehman, and K. Martin, "Host mobility key management in dynamic secure group communication," *Wireless Networks*, vol. 24, pp. 3009–3027, 2018.
- [18] M. Anzani, H. Haj Seyyed Javadi, and V. Modirir, "Key-management scheme for wireless sensor networks based on merging blocks of symmetric design," *Wireless Networks*, vol. 24, pp. 2867–2879, 2018.
- [19] M. A. Al-taha and A. M. Ra'ad, "Lightweight key management scheme for hierarchical wireless sensor networks," in *Proceedings of 7th International Conference on Computer Science, Engineering & Applications (ICCSEA 2017)*. Aircc Publishing Corporation, 2017, pp. 139–147.
- [20] M. S. Yousefpoor and H. Barati, "Dskms: a dynamic smart key management system based on fuzzy logic in wireless sensor networks," *Wireless Networks*, vol. 26, no. 4, pp. 2515–2535, 2020.
- [21] A. Singh, A. K. Awasthi, and K. Singh, "Lightweight multilevel key management scheme for large scale wireless sensor network," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2016, pp. 3014–3017.
- [22] Q. Cheng, C. Hsu, and L. Harn, "Lightweight noninteractive membership authentication and group key establishment for wsns," *Mathematical Problems in Engineering*, vol. 2020, no. 1, p. 1452546, 2020.
- [23] A. Helali, A. Msolli, H. Maaref, and R. Mghaieth, "Kmph: Key management scheme based on pool-hash for wsn," *Journal of Circuits, Systems and Computers*, vol. 30, no. 01, p. 2150003, 2021.
- [24] V. Arumugam, A. Seshasayanam *et al.*, "An adaptive multilevel location based key management system for dynamic wireless sensor networks," *International Journal of Applied Science and Engineering*, vol. 18, no. 1, pp. 1–11, 2021.




- [25] V. Kumar, N. Malik, G. Dhiman, and T. K. Lohani, "Scalable and storage efficient dynamic key management scheme for wireless sensor network," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–11, 2021.
- [26] P. Ahlawat and M. Dave, "Secure path key establishment schemes based on random key management for wsn," *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, vol. 91, pp. 555–567, 2021.
- [27] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

## BIOGRAPHIES OF AUTHORS






**Mohammed Adnan Altaha**    is a Lecturer at university of Basrah. Complete his Bachelor's degree in Computer Science from the College of Education for Pure Science, University of Basrah, Basrah, Iraq (2009), completed his master's degree in Computer Science from the college of Science, University of Basrah, Basrah, Iraq (2018), currently works as a lecture in the College of Veterinary, University of Basrah, Basrah, Iraq, published several scientific researches in computer science. He can be contacted at email: [mohammed.altaha@uobasrah.edu.iq](mailto:mohammed.altaha@uobasrah.edu.iq).






**Wisam Mahmood Lafta**    Was born in Baghdad, Iraq. Received a BSc in computer science from the University of Technology; the MSc at Huazhong University of Science and Technology in China. He is currently a faculty member in the computer science department, University of Technology, Baghdad, Iraq. He has some important published papers in international journals and a reviewer at some international journals. He can be contacted at email: [wisam.m.lafta@uotechnology.edu.iq](mailto:wisam.m.lafta@uotechnology.edu.iq).



**Ghazwan Abdulnabi Al Ali**    received the B.S. degree in Computer Science from Iraq, University of Basra, and the M.S. degree in Computer Science from The University of Science Malaysia. He is currently working as a programmer at the University of Basra. His research interests include software engineering and deep learning. He can be contacted at [ghazwan.alali@uobasrah.edu.iq](mailto:ghazwan.alali@uobasrah.edu.iq).



**Ahmed Adil Alkadhawee**    is a Ass. Profesore at Basrah University, Iraq. He holds an M. Sc degree in Computer Engineering at Huazhong University of Science and Technology in China. He is research areas are Wireless Sensor Network, Machine Learning and Deep Learning. He has authored more than 11 publications: 1 proceeding and 10 journals, with 3 Hindex and more than 19 citations. He can be contacted at email: [ahmedadel@uobasrah.edu.iq](mailto:ahmedadel@uobasrah.edu.iq).